LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-659530

Trend Analysis in Biomedical Texts via Vector Space Model Synonym Extraction

B. Buscaino

September 2, 2014

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

IM Release Number: LLNL-TR-659530

Trend Analysis in Biomedical Texts via Vector Space Model Synonym Extraction

Brandon Buscaino
Lawrence Livermore National Laboratory

Spetember 5, 2014

Abstract

Accurate synonym extraction is a highly desired tool in computational semantics and natural language processing (NLP). Intrinsic to many current language models is the Vector Space Model (VSM) whereby linguistic properties are modeled by real vectors of small length compared to the document size. Here, several vector models are discussed. Utilizing the open source Word2Vec tool, temporal trends in biomedical abstracts from MEDLINE have been analyzed via a unique transformation matrix method. Additionally, accuracy of the word vector method and clustering of temporal trends are discussed.

1 Introduction

Synonym extraction is an extremely useful tool in many aspects of computational linguistics. Semantic similarity can be utilized for simple word translation, identification of syntactic structure, and even for machine-generated query response. These applications allow for a more complete machine-generated representation of language, a large barrier in many natural language processing (NLP) tasks. Current large-scale projects such as WORDNET seek to create synonym-dominated representations of language that incorporate varying degrees of lexical similarity and syntactic structure to create generalized language models [1]. However, generalized models often incorrectly represent words in technical or specialized corpora. Thus, to ensure accuracy in large but highly specific texts, focused language-representation models are needed to prioritize the unique vocabulary of these corpora.

2 Vector Space Model

Recently, Vector Space Models (VSMs) have been effectively used for automatic synonym extraction with comparable results to WORDNET [2]. VSMs are algebraic models of language that utilize high-dimensional vectors to reproduce semantic similarity, first introduced by [3]. Depending on the size of the corpus, size of the vocabulary and desired accuracy in reproduction, vector dimensionality can be optimized for performance. All VSM models rely on corpus-specific text analysis to create vectors that represent a context by which a term is identified. This approach is effective as it more accurately reflects human language usage. In the same way that humans can intuit the meaning of infrequently used words by their context, VSMs derive word similarity from context.

For the most basic co-occurrence VSM, the target corpus is parsed into documents (which can be sentences, paragraphs, or, in some cases, actual large documents). Each document is assigned a unique document vector in V-dimensional space where V is the number of total vocabulary words. Each document vector is composed of a normalized count of each unique word within the document, as each dimension in V represents a different vocabulary word. Thus, a V x d co-occurrence matrix is formed, with d as the number of documents. This set of d-dimensional word vectors (viewing, instead, each word vector as a representation of the documents that describe it) then undergoes a dimensional reduction from d to k where $k << d$ and usually $k \sim 100$ [4]. Working in this significantly smaller space, word similarity can be calculated using a simple cosine distance measure [5]:

$$sim(\vec{A},\vec{B}) = \frac{\vec{A}\cdot\vec{B}}{||\vec{A}||\ ||\vec{B}||} \tag{1}$$

In this way, words can be compared to one another to provide relevant query retrieval within large corpora. This model supports document retrieval (the original word vectors tell the user which documents the term also occurs in) and also synonym extraction by comparing the similarity of word vectors. A necessary addition to this algorithm is the introduction of a term frequency weighting for each element in the document vector. For nearly all purposes, such as recall, common words such as "the", "or", and "and", which retain little semantic meaning, should be ignored. Conversely, words that occur rarely in the corpus should not be given a vector representation due to the lack of context required to accurately model words. To weight terms properly, a scheme titled Term Frequency-Inverse Document Frequency (TF-IDF) is often implemented to reward terms that occur frequently within a document but are not represented in all documents. TF-IDF weighting can be calculated for a term in a document by the following equations [6]:

$$TFIDF = TF * IDF \tag{2}$$

$$IDF = \log\frac{d}{df} \tag{3}$$

Where TF is the number of times that the term occurs in the document, d is the number of total documents and df is the number of documents in which the word is found. Nearly all VSMs contain term weighting to remove words with no contextual meaning. Note that the simple co-occurrence model described here is also called a "bag-of-words" (BOW) model where each document is treated as a collection of words with no syntactic structure preserved. An in-depth review of semantic space models can be found in [7].

3 Word Vector Models

3.1 Random Indexing

Many current vector models run into computational complexity issues when analyzing large corpora due to the creation and manipulation of the document-word co-occurrence matrix. Since the matrix entry number scales with V $*$ d, texts composed of many documents require difficult matrix manipulation. Random Indexing (RI) removes this problem by exploiting the near-orthogonality of random, sparse, and high-dimensional vectors. The initial co-occurrence matrix is bypassed by representing each document as a sparse k-dimensional random vector (k $\sim$ 100). Word vectors are initialized as empty vectors and for every word in a document, the document vector is added to the word vector. This produces a k x V dimensional matrix where each word is a sum of its "contexts", or in other words, documents. Random Indexing can often be extended to run in conjunction with different algorithms as it is largely a dimensionality reduction technique [8]. Research reports that run-time can be halved by incorporating random indexing techniques [9].

3.2 Skip-gram

N-grams, contiguous sequences of n characters (in this context, words), are an important aspect of language modeling as they can represent common phrases in text. One issue with n-gram implementation is the data sparsity problem, or that "language is a system of rare events, so varied and complex, that we can never model all possibilitie" [10]. Most languages, then, contain few meaningful contextual words, and are filled with rather unimportant filler words. In order to account for this, the skip-gram model is introduced. Given a document of n terms, the skip-gram model seeks an m-gram (m < n) that will provide context to each word. Different from the BOW model, skip-gram classifies words based off the words before and after the target word. These m-words can be split up from the total n words in different permutations to increase precision [11]. The skip-gram modeling technique is the model used for this project.

3.3 Latent Semantic Analysis/Latent Dirichlet Allocation

Most vector models (RI as an exception) require dimensionality reduction techniques. From a weighted co-occurrence matrix, how can the dimensionality be reduced such that retained semantic information is maximized? Latent Semantic Analysis (LSA) reduces the dimensionality of the rows, corresponding to documents, by performing a singular value decomposition (SVD) on a related matrix to maximize the L2 norm of the word vectors [12]. Latent Dirichlet Allocation (LDA), another dimensionality technique, classifies documents based on latent topics. A document, for example, is then represented as a weighted sum of the topics. In turn, each topic is composed of a distribution of words. When dimensionality is reduced, the model maximizes the preservation of topic structure [13]. Both of these models outperform basic models, but are computationally expensive.

3.4 Current Model

In this paper, the open source library, Gensim [14], which implements the open source tool Word2Vec [15], was used. The Word2Vec tool was implemented from algorithms developed in [11]. The Gensim library supports model training and word vector comparison tools such as cosine similarity. Additionally, Gensim offers the ability to efficiently find the top N related terms to a query based on a trained model. For this project, the skip-gram implementation was utilized with varying degrees of dimensionality to understand temporal trends in a corpus of biomedical abstracts taken from the MEDLINE database [16].

4 Trends in Biomedical Data

Language modeling in biomedical texts has a long history [17, 18]. Biomedical texts are often difficult to model due to large technical vocabularies, a multitude of research topics and high variability in scientific naming practices. Thus, efficient and effective ways to highlight research trends are necessary. In this paper, focus is placed on analyzing temporal trends in biomedical literature using a VSM to measure which words have changed the most. In this context a word has changed over time if its constituent word vector has changed with relation to other word vectors. In order to analyze temporal variation in word vectors, both a reference model and a current model are necessary (often referred to as first/second or before/after).

5 Temporal Trend-Finding Methods

5.1 Perturbation Method

The perturbation method utilizes the built in functions of Gensim to analyze word vector evolution. First a model is trained on all data up until an arbitrary user-input date. The model is saved and then training is continued on all available data after the given date in order to create the current model. This method's largest benefit is its computational cost. The Word2Vec tool is highly optimized for performance and training on new data is cheap. No additional information is needed to understand the change between the word vectors. Comparison between words in each model is a task of computing the cosine similarity of words in both models.

However, this model has the drawback that the second data set must be small in relation to the first. This is required because the vector basis of the first model cannot change significantly; otherwise it loses its comparative power. In other words, training can only occur on the model so long as each vector dimension represents the same information. When training is continued on a larger set of data, each dimension loses its original meaning and comparing word vectors becomes more difficult. The semantic information is still there, but more analysis is necessary to extract the information. Specifically, a transformation is necessary to translate between the 2 spaces.

5.2 Transformation Matrix Method

A more robust method that does not require that the new training set be small is a transformation matrix method. In this method, two VSMs are created for both the first and second data set. Once the models are trained, the intersection of their vocabularies is taken and the model has then produced two n x V matrices where n is an optimized number of dimensions ($n \sim 100$). However, since the bases of each vector set are different between models, a transformation must be taken to map the two spaces. Since this problem is overdetermined (V > n), a least squares algorithm is necessary to minimize the distance between transformed word vectors.

Early versions of this method implemented the transformation-finding algorithm of [19], but proved too computationally expensive for large dimensions. A more appropriate model is an efficient least-squares algorithm based off of the singular value decomposition (SVD) [20]. This model is robust as the SVD is an efficient and accurate decomposition tool. The output of this method is a transformation matrix (technically a rotation matrix with a translation vector) that maps between the basis sets of both models.

Unfortunately, this model can over-represent sparse words. Words that have been used many times in the first data set have a fixed context. However, if that same word is used infrequently in the second data set, the model will have a highly variable vector representation for that word. It is likely, then, that when the transformation matrix is applied to the initial word vector, the resultant vector will deviate largely from the word vector in the second model. Thus, the method would determine that the word had changed significantly when, in reality, there was simply a lack of context. This effect was controlled by creating a threshold usage parameter. The model in this paper required that a word be used at least 10 times, though an analysis of optimal threshold usage parameters was not performed.

6 Trend Analysis

6.1 Accuracy of Word Vectors/Transformation Matrix

Before considering actual trend analysis, it is prudent to check the accuracy of the word vectors at representing a complete set of semantic knowledge of a corpus. Dimensionality reduction necessitates the loss of information in all but the simplest cases. Thus, quantification of lost information is an essential task. Often, researchers analyze word accuracy by comparing word vector predictions to human tests [21]. In fact, Word2Vec developers report over 70% accuracy in human synonym extraction tests [15]. However, this paper proposes that word vector information storage can be measured via the following method.

First, train two Word2Vec models on the same data set. However, train the second model data in a different order. Since the model treats each individual sentence as unique documents, the information retained by each model is identical. Yet, since the data is trained in a different order, the constituent basis of each model will be different. Truly, they will be different even when data is trained in the same order due to randomness within the dimensionality reduction techniques, but this is a cautious step to ensure the vector spaces are distinct. Mapping similar spaces to each other can be a challenging task, especially with regards to machine precision considerations.

Then, a transformation matrix is found between the two sets of data given by the SVD algorithm described above. Once the transformation matrix is computed, each vector in the first model is mapped to the basis of the second model. Then 1000 samples of the transformed vectors from the first model are compared to the entire second model to find the best cosine similarity fit. If the best-fit word in the second model is identical to the initial word in the first model, the event is recorded a successful reproduction. Note that the number of samples was determined by run-time limitations as opposed to an alternative optimization. An ideal implementation would test the entire intersection of the vocabularies.

This test addresses the loss of information due to dimensionality reduction in the word vector model. Figure 1 demonstrates a test on different size data sets. As expected, for any size model, increasing the vector dimensionality increases the percent reproduction. Interestingly, larger data sets perform better than smaller ones although they contain a larger vocabulary. This effect is explained by the fact that the vocabulary does not scale linearly with corpus size. So, in the larger model, each word has more context than the smaller model, allowing the vector tool to create a more accurate representation of the word. As shown in Figure 1, vectors models with low dimensionality behave poorly and are wildly inaccurate. Larger data sets cannot improve accuracy in this regime due to the hard limit of information storage in low-dimensional vectors. It is evident that word vectors of approximately 100 dimensions for these data sets are sufficient for accurate recall and precision.

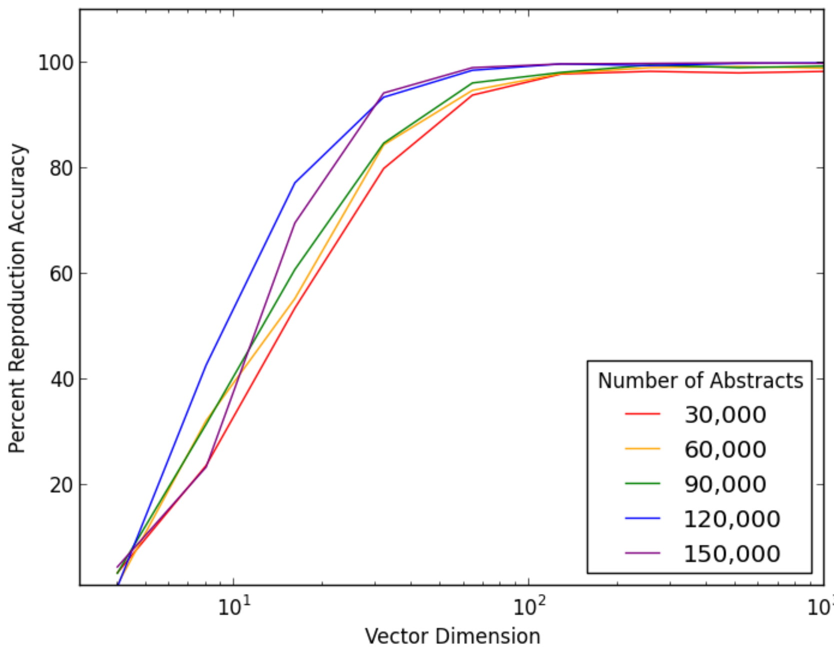

Figure 1: Reproduction accuracy compared to vector dimensionality for different size corpora. Notice that larger texts perform better in smaller dimensions. This occurs because the vocabulary size increases logarithmically with corpus size while the context for each word increases linearly. The context-to-word ratio increases and thus each vector dimension is able to more precisely model semantic information

6.2 Word Evolution Analysis

A list of the most-changed words for a given split date using the algorithm described in the Transformation Matrix Method section is shown in Table 6.2. For example, the most changed word since 2012 is "huji", which is the abbreviation for the Hebrew University of Jerusalem. Interestingly, universities tend to be over-represented in the more recent data. This trend can be easily explained by the dynamic nature of university research. Universities often have projects that come and go quickly, depending on funding from different sources. While finding a suitable metric for change is difficult, this list provides many reasonable answers. In fact, it appears that this method is able to find rising research topics such as the inhibitor DAPT, the symptom Migraine Without Headache (MAWOH), and even statistical methods used in biological research such as Tract-based Spatial Statistics (TBSS).

However, it is clear that there is noise present in this data. For example, words such as "noninferior" and "7sr0" do not contain relevant semantic meaning for this corpus. A more involved analysis could weight terms according to a TFIDF scheme to ensure relevant results, though such a method was not implemented here.

Rank	1990	2000	2004	2008	2012
1	fhxs	dapt	tbss	affrc	huji
2	shh	mawoh	affrc	kobic	cochranemsk
3	arbs	sirt	reltx	uniroma2	unige
4	wst	embryonization	univie	cnptia	swmed
5	embryonization	weinicom	dapt	univie	oupjournals
6	grn	dnmts	ncifcrf	ncifcrf	circresaha
7	sscp	ngs	cfald	linq	ernet
8	xpert	noninferior	nursingmanagement	dpseh	kyutech
9	esl	splenotomy	autarcesis	7sr0	cebitec
10	cdn	t2d	oif	nursingmanagement	aporc

Table 1: Top 10 most-changed words for different time periods. Each date corresponds to the cutoff date for the corpus. For example, the 2000 column correspond to the most-changed words between the time periods 1950 - 2000 and 2000 - present.

6.3 Clustering Analysis

While the list structure above is useful in determining specific words that have evolved over time, it fails to find overarching themes in the data. This well-known problem falls under the domain of clustering analysis. For this time-limited project, an invested clustering algorithm was not developed, but rather, a visualization scheme was utilized. Given a set of 100-dimensional vectors, the most-changed words were visualized through a dimensional reduction technique that has been shown to preserve semantic relations [22]. In this paper, I use the Multi-Dimensional Scaling (MDS) and Principal Component Analysis (PCA) tools from the open source library Scikit-learn [24]. MDS has shown to be effective in other semantic tasks such as opinion mining [25].

Figure 2 visualizes the set of most-changed words before (blue) and after (green) the split date. This was calculated by mapping the first model vectors for the 1000 most-changed words since 2012 to the second models subspace using the transformation matrix method. Then the MDS algorithm was run on both sets of data and the points plotted. Figure 3 implements the same method, but uses Principal Component Analysis (PCA) for the dimensional reduction as a check of the accuracy of the 2D vectors. It is clear that the dimension reduction algorithms map to different point structures and comparison of Figures 2 and 3 indicates that PCA prefers point positive mappings while MDS maps the vectors to a "bullseye" structure.

Additionally, it is necessary that the MDS tool be run on both sets of vectors at the same time instead of independently. In the latter case, an effect like that observed in Figure 4 is observed (for MDS), whereby an apparent shift in data is observed. This is an artifact of the lack of information connection between the basis vectors. The MDS algorithm will map the first set of vectors to one space, while the second to a different space; plotting them on the same graph has no meaning.

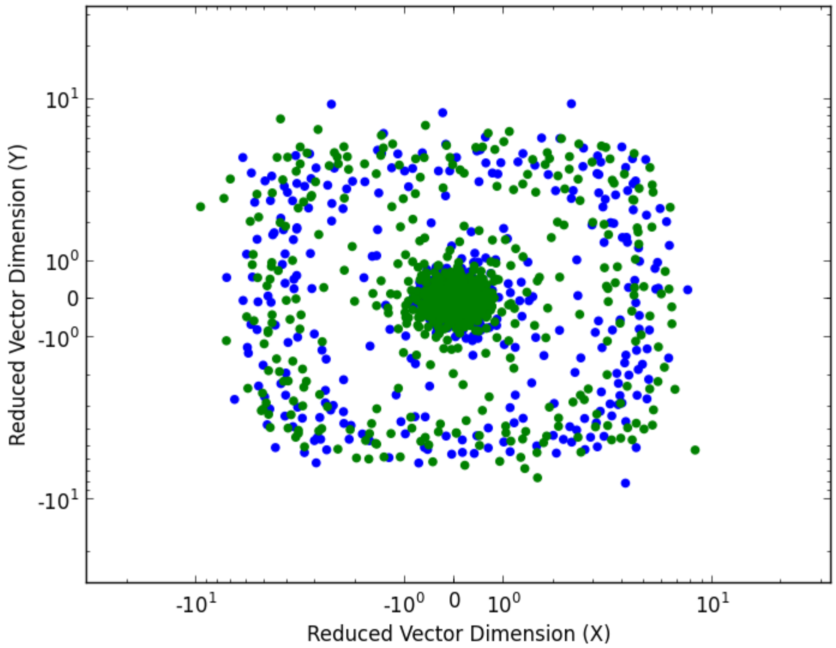

Figure 2: 1000 most-changed words before and after 2012. They have been reduced to two dimensions using the MDS reduction algorithm.

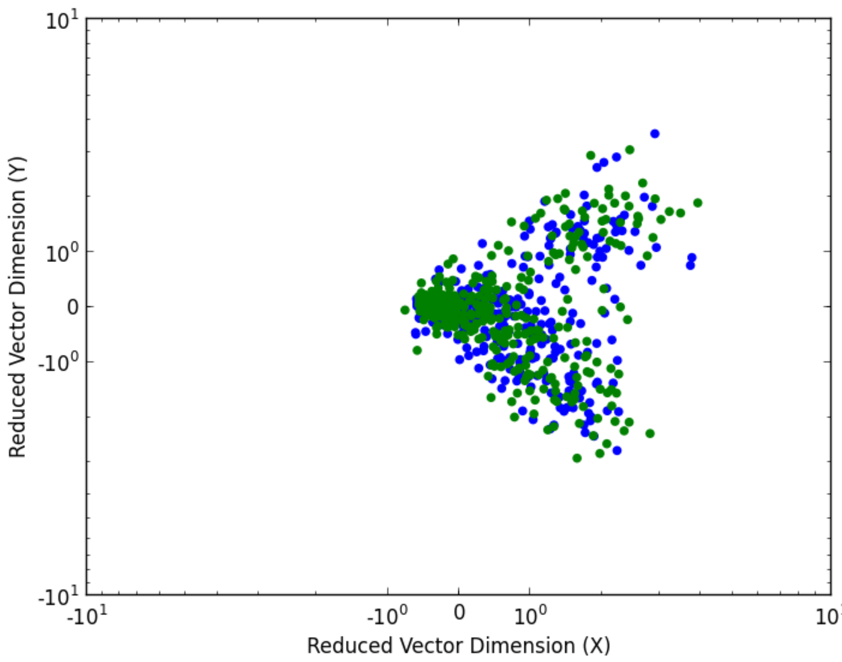

Figure 3: 1000 most-changed words before and after 2012. They have been reduced to two dimensions using the PCA reduction algorithm.

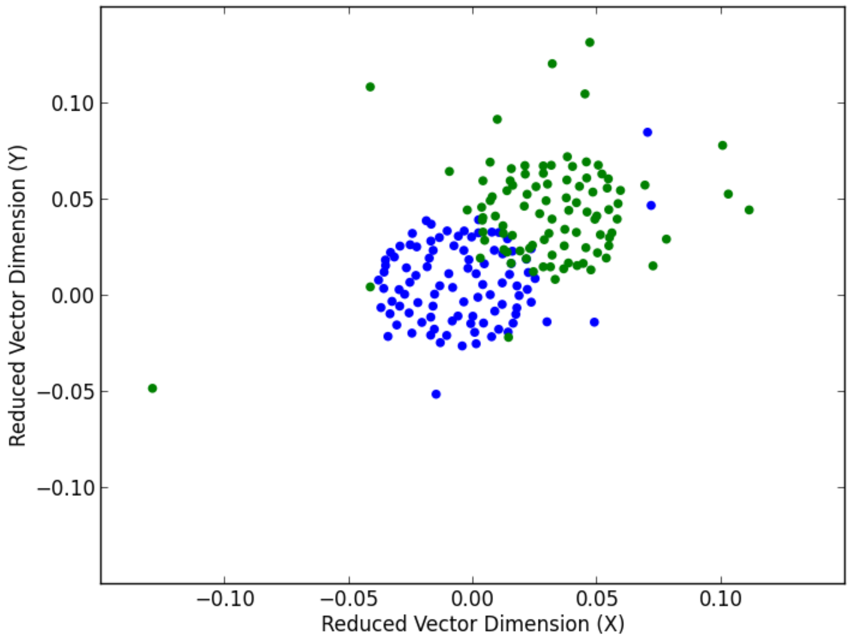

Figure 4: Effect of independently reducing word vectors. A comparison between these points is useless as the reduced dimensions for each set have different meanings.

However, this visualization scheme fails to capture point-to-point mappings for individual words. A more beneficial model first finds the top 1000 most-changed 100 dimensional word vectors, subtracts the first and second model vectors, maps this difference to 2 dimensions, and then plots them. In this way, the most relational information can be retained. This is shown in Figures 5 and 6 for MDS and PCA algorithms respectively. Notice that the point structure for the MDS algorithm is similar to previous visualization methods, indicating that MDS prefers specific point distributions, a trait that is problematic for analyzing point structure after dimensional reduction.

In these figures, the red points represent the 50 most-changed words in model. Notice that they are located nearly exactly at the center, whereas words that differ from each other are expected to deviate from the origin significantly. Again, this is explained as an artifact of the dimensional reduction techniques. A promising result, however, is that all of the most-changed words are preferentially mapped to the center of the plot. This indicates that the dimension reduction algorithm knows which words have changed the most and does not distribute them randomly about the origin as occurs with the other words.

7 Future Work

It is important to note that the tests above only represent tests for unigrams (single words). However, language is quite variable in its representation of unique concepts, often taking 3 or more words to represent a single object or idea. Thus, an extension of this project to represent phrases, potentially through the Word2Phrase tool [15], could offer more insight into temporal trends, or at the very least, increase accuracy.

Additionally, it has been shown that transformation matrix methods can be used for machine translation through a similar process to the one described in this paper[22]. Further research into the accuracy of these methods could provide insight into the problems encountered in this paper.

Finally, the clustering methods presented in this paper offer only a visual representation of word vector clustering. A more involved method might use the common K-means clustering algorithm to determine larger temporal research trends. Investigation of the accuracy of MDS and PCA in

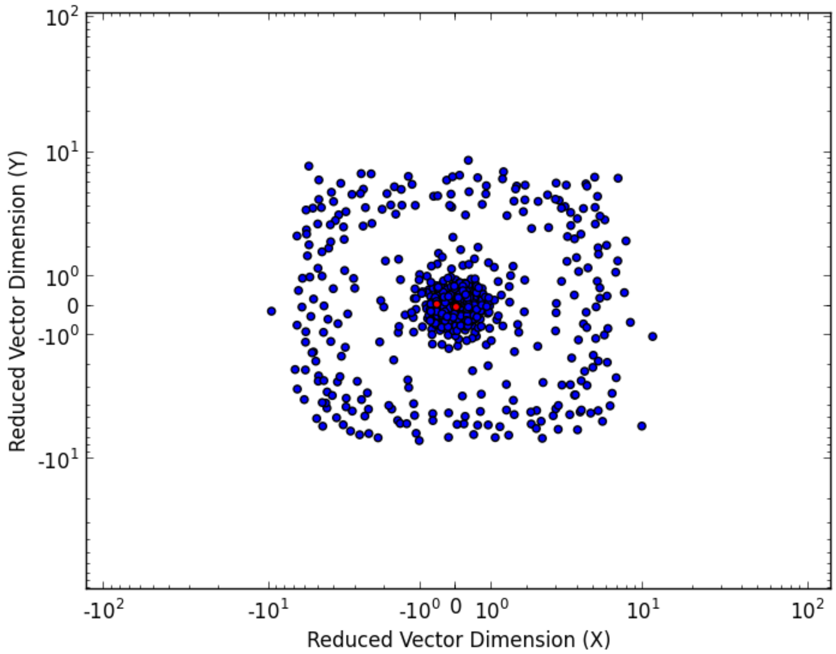

Figure 5: Word vector changes mapped to two dimensions using the MDS algorithm.

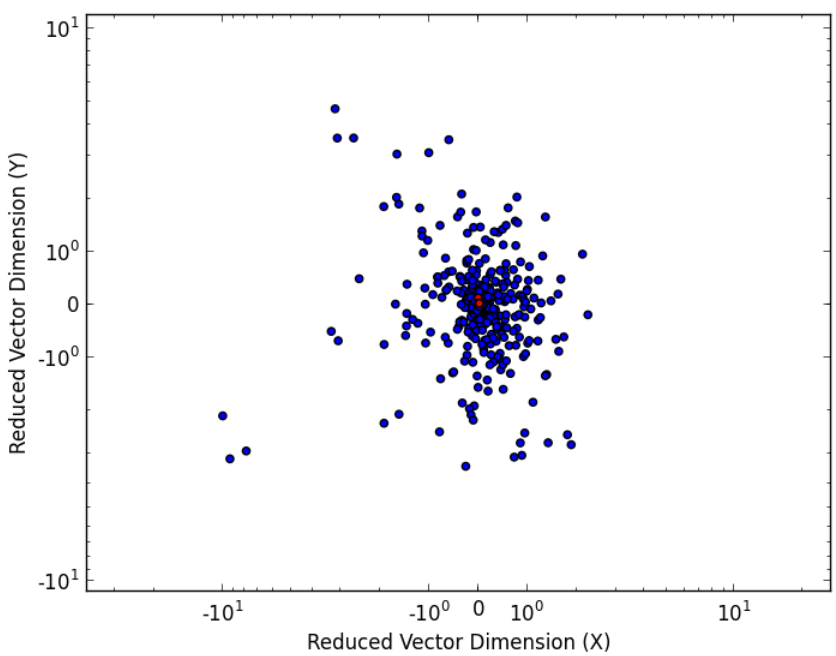

Figure 6: Word vector changes mapped to two dimensions using the PCA algorithm.

preserving semantic word distance is also necessary.

8 Conclusion

An introduction to several common vector space modeling techniques is presented and a unique transformation matrix method is used to map vector spaces created by the Word2Vec tool. It is apparent that 100 dimensional vectors are sufficiently large enough to contain nearly all semantic information from our corpus of 150,000 MEDLINE abstracts, corresponding to approximately 20 million words. The method developed for word evolution analysis shows promising results at reproducing past biomedical trends. Clustering analysis for these models is performed with the MDS and PCA dimensional reduction algorithms, resulting in interesting, but somewhat unhelpful point structures in lower dimensions.

9 Acknowledgements

I would like to thank my mentor Tim Bender, Jeff Drocco, and the entire BKC team for their help, advice, and guidance in this project. This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).

References

[1] Pedersen T., Patwardhan S., and J. Michelizzi, 2004. "WordNet::Similarity: measuring the relatedness of concepts." In Demonstration Papers at HLT-NAACL 2004 (HLT-NAACL–Demonstrations '04). *Association for Computational Linguistics*, Stroudsburg, PA, USA, 38-41.

[2] Hagiwara M., Ogawa Y., and Toyama K. "Selection of Effective Contextual Information for Automatic Synonym Acquisition." Proc. COLING/ACL (2006) 353360.

[3] Salton G., Wong A., and Yang C. S. 1975. "A vector space model for automatic indexing." Commun. ACM 18, 11 (November 1975), 613-620. DOI=10.1145/361219.361220. http://doi.acm.org/10.1145/361219.361220.

[4] Sahlgren, M. (2005). "An introduction to random indexing." In Proceedings of the Methods and Applications of Semantic Indexing Workshop at the *7th International Conference on Terminology and Knowledge Engineering* (TKE), Copenhagen, Denmark

[5] Karypis G. and Han E. 2000. "Fast supervised dimensionality reduction algorithm with applications to document categorization & retrieval." In Proceedings of the *Ninth International Conference on Information and Knowledge Management* (CIKM '00). ACM, New York, NY, USA, 12-19. DOI=10.1145/354756.354772. http://doi.acm.org/10.1145/354756.354772.

[6] Manning, C. D., Raghavan, P., Schutze, H. (2008). "Scoring, term weighting, and the vector space model." *Introduction to Information Retrieval*. p. 100. doi:10.1017/CBO9780511809071.007.

[7] Pad S. and Lapata M. 2007. "Dependency-Based Construction of Semantic Space Models." *Comput. Linguist.* 33, 2 (June 2007), 161-199. DOI=10.1162/coli.2007.33.2.161 http://dx.doi.org/10.1162/coli.2007.33.2.161.

[8] Henriksson A., Moen H., Skeppstedt M., Daudaravi V., Duneld M., et al. 2014. "Synonym extraction and abbreviation expansion with ensembles of semantic spaces." Journal of Biomedical Semantics, 5(1):6.

[9] Sellberg L. and Jnsson A. 2008. "Using random indexing to improve singular value decomposition for latent semantic analysis." In Proceedings of the *Sixth International Conference on Language Resources and Evaluation* (LREC'08), Marrakech, Morocco.

[10] Guthrie, D., Allison, B., Liu, W., Guthrie, L., & Wilks, Y. (2006). "A closer look at skip-gram modelling." *In Proceedings of the Fifth International Conference on Language Resources and Evaluation* (LREC-2006) (pp. 12221225).

[11] Mikolov T., Chen K., Corrado G., and Dean J. "Efficient estimation of word representations in vector space." ICLR Workshop, 2013.

[12] Dumais, S. T. (2004), “Latent semantic analysis.” Ann. Rev. Info. Sci. Tech., 38: 188230. doi: 10.1002/aris.1440380105

[13] Blei D., Ng A., and Jordan M. 2003. “Latent dirichlet allocation.” J. Mach. Learn. Res. 3 (March 2003), 993-1022.

[14] Rehurek R. and Sojka P. “Software Framework for Topic Modelling with Large Corpora.” ELRA. p. 45–50 2010.

[15] https://code.google.com/p/word2vec/

[16] http://www.nlm.nih.gov/bsd/pmresources.html

[17] Cohen A, Hersh W, Dubay C, Spackman K. “Using co-occurrence network structure to extract synonymous gene and protein names from MEDLINE abstracts.” BMC Bioinformatics 2005, 6:103.

[18] Masseroli, M., Chicco, D., and Pinoli, P. “Probabilistic Latent Semantic Analysis for prediction of Gene Ontology annotations,” , *The 2012 International Joint Conference on Neural Networks* (IJCNN), June 2012. doi: 10.1109/IJCNN.2012.6252767

[19] Spath H. “Fitting affine and orthogonal transformations between two sets of points.” Mathematical Communications, 9:2734, 2004.

[20] Sorkine, O., 2007. “Least-squares rigid motion using svd.”

[21] Pennington J., Socher R. and Manning C. “Glove: Global Vectors for Word Representation.” *Conference on Empirical Methods in Natural Language Processing* (EMNLP 2014)

[22] Mikolov T., Le Q., and Sutskever I. “Exploiting Similarities among Languages for Machine Translation.” Technical report, arXiv, 2013.

[23] Jaworska N, Chupetlovska-Anastasova A. “A review of multidimensional scaling (MDS) and its utility in various psychological domains.” Tutorials in Quantitative Methods for Psychology. 2009;5:110.

[24] Pedregosa et al. Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

[25] Cambria E, Song Y, Wang H, Howard N (2013) “Semantic multi-dimensional scaling for open-domain sentiment analysis.” IEEE Intell Syst. doi:10.1109/MIS.2012.118.